Amendments to the Specification:

Please amend the section entitled "Brief Description of the Drawings" (beginning with the paragraph on Page 2, Line 28) with the following:

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

FIG 1 is a block diagram of a data processing system;

FIG 2 is a block diagram illustrating selected features of a processor suitable for use in the data processing system of FIG 1;

FIG 3 is [an exemplary] a prior art code segment [suitable for implementing the] that might benefit from the use of the conditional branch information of the present invention;

FIG 4 illustrates a branch conditional statement according to the present invention; and

FIG 5 is a block [digram] diagram illustrating features of a [branchprediction] branch prediction unit according to one [embodiment] embodiment of the present invention.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description presented herein are not intended to limit the invention to the particular embodiment disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

Please amend the paragraph beginning on line 15, page 8, as follows:

The prediction may be based on prior executions of the same conditional branch statement using an instruction history table that records the [results] conditional branch instruction results. Branch prediction may also be improved by incorporating prediction information into the branch instruction itself. In this approach, the compiler evaluates the context in which a conditional branch statement is executed and makes a determination, if possible, about whether the branch is likely to be taken. The conditional branch statement at the end of a loop that is executed 100 times, for example, branches to the same instruction address 99% of the time. In this case, the compiler could embed information in the branch instruction itself (assuming there are bits positions available in the instruction) to tell the hardware the direction in which way the branch is most likely to go.

Please amend the paragraph beginning on line 13, page 10, as follows:

Referring momentarily to FIG 4, a conditional branch instruction formatted in accordance with the PowerPC® instruction set and suitable for use with the present invention is presented. In the illustrated example, branch conditional (BC) instruction 400 includes a 6-bit primary opcode field 402, and a 5-bit secondary opcode field [404] 404. Secondary opcode field 404 indicates whether the branch is taken if the appropriate condition (represented by a bit in the condition register) is true or false. In addition, the depicted embodiment of secondary opcode field 404 includes a pair of branch prediction information bits, identified as the XY bits. The XY bits are generated during compilation of the executable code based upon the compiler's interpretation of the context in which the conditional branch statement is used. If the compiler determines that branch prediction is unlikely to significantly improve the branch prediction rate, the compiler can encode the XY bits with an appropriate value to indicate that branch prediction is to be bypassed. Similarly, if the compiler determines from its context that a particular branch conditional instruction is susceptible to accurate branch prediction, the compiler can encode the XY bits of the branch instruction accordingly.